



**Національний технічний університет України  
“Київський політехнічний інститут”**

**Факультет прикладної математики  
Кафедра системного програмування і спеціалізованих  
комп’ютерних систем**

**Лабораторна робота №3  
з дисципліни “Комп’ютерна графіка”**

*Тема: “Алгоритми заливки заданих контурів (областей)”*

Виконав:  
студент III курсу  
групи КВ-01  
Таранич Артем

**Тема:** Алгоритми заливки заданих контурів (областей)

**Мета роботи:** на практичному досвіді ознайомитись із способами заливки довільно заданих контурів (областей) на екрані монітору.

**Постановка задачі:** Програмно реалізувати 4 різні алгоритми заливки (на вибір) довільно заданих областей на екрані монітору.

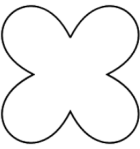

**Завдання для лабораторної роботи:**

1. Задати області екрану двох видів (многокутник та довільний контур).
2. Заповнити задані області деяким кольором.
3. Порівняти реалізовані алгоритми за швидкістю (на прикладі заповнення одного із контурів) .

**Код доступный на githabi:**

[https://github.com/Unberant/ComputerGraphics/tree/lab\\_3](https://github.com/Unberant/ComputerGraphics/tree/lab_3)

Результати



ПОЧАТКОВИЙ СТАН

АЛГОРИТМ ЗАТРАВКИ

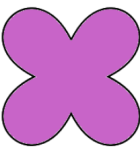
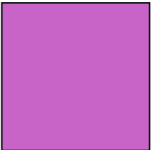
АЛГОРИТМ ХОР

РЕКУРСИВНИЙ АЛГОРИТМ

ЧОТИРЬОХЗВ'ЯЗНИЙ АЛГОРИТМ

Витрачений час для 1 фігури:

Витрачений час для 2 фігури:



ПОЧАТКОВИЙ СТАН

АЛГОРИТМ ЗАТРАВКИ

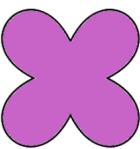

АЛГОРИТМ ХОР

РЕКУРСИВНИЙ АЛГОРИТМ

ЧОТИРЬОХЗВ'ЯЗНИЙ АЛГОРИТМ

Витрачений час для 1 фігури: 668 мілісекунд

Витрачений час для 2 фігури: 454.8 мілісекунд



ПОЧАТКОВИЙ СТАН

АЛГОРИТМ ЗАТРАВКИ


АЛГОРИТМ ХОР

РЕКУРСИВНИЙ АЛГОРИТМ

ЧОТИРЬОХЗВ'ЯЗНИЙ АЛГОРИТМ

Витрачений час для 1 фігури: 547.2 мілісекунд

Витрачений час для 2 фігури: 375.9 мілісекунд



ПОЧАТКОВИЙ СТАН

АЛГОРИТМ ЗАТРАВКИ


АЛГОРИТМ ХОР

РЕКУРСИВНИЙ АЛГОРИТМ

ЧОТИРЬОХЗВ'ЯЗНИЙ АЛГОРИТМ

Витрачений час для 1 фігури: 460.1 мілісекунд

Витрачений час для 2 фігури: 284.6 мілісекунд



ПОЧАТКОВИЙ СТАН

АЛГОРИТМ ЗАТРАВКИ

АЛГОРИТМ ХОР

РЕКУРСИВНИЙ АЛГОРИТМ

ЧОТИРЬОХЗВ'ЯЗНИЙ АЛГОРИТМ

Витрачений час для 1 фігури: 573.3 мілісекунд

Витрачений час для 2 фігури: 401.4 мілісекунд

## Порівняння алгоритмів

Для порівняння було використано результати роботи алгоритмів на заливці прямокутного контуру (оскільки для другого контуру, утвореного кривою Безьє, XOR має свої особливості).

За результатами виміру швидкодії алгоритмів протягом 15 дослідів було сформовано таблицю 1.

Таблиця 1. Виміри часу роботи (в наносекундах) кожного з розглянутих алгоритмів протягом 15 дослідів та середнє значення швидкодії алгоритмів.

№ досліду	Алгоритм			
	Рекурсивний	Заливка	Чотирьохзв'язний	XOR
1	465.3	676.1	569.2	553.8
2	474.1	686.7	589.3	547
3	466	655.1	567.4	554
4	459.5	682.5	571.4	610.3
5	461.5	668.9	583.9	588.5
6	465.6	689.3	578.5	567.9
7	458.4	677.9	573.6	594.5
8	445	666.3	610.6	574.3
9	458.6	671.5	573.6	573.1
10	460.5	680.8	576	577.9
11	460.3	666.9	575.2	592.9
12	464.9	665	583.6	573.8
13	466.4	722.2	569.2	594.4
14	469.1	665	577.7	578
15	465.3	662.3	569.2	574.1

### Середнє значення часу

462.7	675.77	577.89	576.97
-------	--------	--------	--------

## Висновок

З таблиці 1 отримуємо дані, що XOR виявився найдовшим в своїй роботі та найгіршим у швидкодії, хоча різниця відсутня у візуальному плані. Трапилось це через те, що даний алгоритм заповнює область справа від контуру заливки, що ми бачимо в вигляді дещо більшого за інші відходу від контуру фігури. Для заливки контуру була потрібна умова належності точки контуру, в протилежному випадку розпізнавання не відбувалося. Пояснюватись це може тим, що через заливку з правої сторони ніби стирається колір відповідного контуру і перестає його бачити.

Наступні 2 алгоритми є дуже близькими за швидкістю тому що їх принцип роботи полягає у заповненні точки, а за нею тих, що її оточують. Головну різницю створює той факт, що чотирьохзв'язний алгоритм в цьому випадку включає в себе стек.

У швидкодії переможцем виявився рекурсивний алгоритм, він найкраще заповнює прямокутну область, що й створює основну різницю в часі. Працює він шляхом заливки по рядках, що і надає йому перевагу в прямокутнику. Через нестандартність іншої фігури ця особливість змушує малювати заливку як для лівої/верхньої так і для нижньої/правої областей. Звідси слідує висновок, що даний алгоритм треба використовувати пріоритетно при роботі з прямокутниками, адже це його найбільша перевага.